CLAIMS

Sub B4

1. A processor having a changeable architected state, comprising:

instruction memory for storing instructions;

an instruction pipeline, wherein an instruction which passes entirely through the pipeline alters the architected state and wherein the pipeline comprises circuitry for fetching instructions from the instruction memory into the pipeline;

circuitry for storing an annul code corresponding to instructions in the pipeline; and

circuitry for preventing one or more selected instructions in the group from altering the architected state in response to the annul code.

2. The processor of claim 1:

wherein the instruction pipeline further comprises a plurality of execution units; and

wherein one of more the plurality of execution units receives a corresponding instruction for executing the corresponding instruction in a given clock cycle.

3. The processor of claim 2:

wherein the annul code comprises a plurality of bit states;

wherein the circuitry for preventing one or more selected instructions in the group from altering the architected state comprises circuitry for coupling the plurality of bit states to respective ones of the plurality of execution units;

wherein in response to a bit state being a first state the execution unit to which the bit state is coupled does not execute the corresponding instruction in the given clock cycle; and

wherein in response to a bit state being a second state different than the first state the execution unit to which the bit state is coupled does execute the corresponding instruction in the given clock cycle.

4. The processor of claim 3 wherein the plurality of execution units comprises a load/store unit, a multiply unit, an ALU unit, and a shift unit.

5. The processor of claim 3:

wherein the plurality of execution units are operable such that in a given clock cycle an integer number N of the plurality of execution units are scheduled to execute; and

wherein the circuitry for coupling the plurality of bit states to respective ones of the plurality of execution units comprises circuitry for coupling only the integer number N of the plurality of bit states to the plurality of execution units which are scheduled to execute in the given clock cycle.

136

6. The processor of claim 5:

wherein the instructions corresponding to the annul code comprise:

a first group of one or more instructions logically arranged after a conditional instruction and to be executed if the condition is satisfied;

a second group of one or more instructions logically arranged after the conditional instruction and to be executed if the condition is not satisfied;

wherein the circuitry for preventing prevents the first group of instructions from altering the architected state in response to the annul code if the condition is not satisfied; and

wherein the circuitry for preventing prevents the second group of instructions from altering the architected state in response to the annul code if the condition is satisfied.

7. The processor of claim 5:

wherein the instructions corresponding to the annul code comprise instructions corresponding to a software loop scheduled to execute for an integer M number of iterations; and

wherein during a given iteration the circuitry for preventing prevents one or more of the instructions corresponding to the annul code from altering the architected state in

response to the annul code and based on a relationship of the given iteration to the integer M number of iterations.

8. The processor of claim 3:

wherein the instructions corresponding to the annul code comprise instructions corresponding to software loop scheduled to execute for an integer M number of iterations; and

wherein during a given iteration the circuitry for preventing prevents one or more of the instructions corresponding to the annul code from altering the architected state in response to the annul code and based on a relationship of the given iteration to the integer M number of iterations.

B4

9. The processor of claim 3:

wherein the group of instructions corresponding to the annul code comprise:

a first group of one or more instructions logically arranged after a conditional instruction and to be executed if the condition is satisfied;

a second group of one or more instructions logically arranged after the conditional instruction and to be executed if the condition is not satisfied;

wherein the bit states corresponding to the first group are set to the first state and the bit states corresponding to the second group are set to the second state if the condition is not satisfied; and

wherein the bit states corresponding to the first group are set to the second state and the bit states corresponding to the second group are set to the first state if the condition is satisfied.

10. The processor of claim 1 wherein the annul code is generated in response to one or more constant generating instructions.

11. The processor of claim 1 wherein the annul code is loaded from a memory.

12. The processor of claim 1 wherein the annul code is an immediate value in an instruction passing through the pipeline.

13. The processor of claim 1 wherein the annul code comprises 32 bits.

14. The processor of claim 1 wherein the annul code comprises more than 32 bits.

36

15. The processor of claim 1:

wherein the annul code comprises 64 bits; and

wherein the annul code is formed in response to two thirty-two bit values.

16. The processor of claim 1:

wherein the annul code is loaded in response to an instruction having a condition predicate;

wherein the annul code comprises a first annul code in response to the condition predicate being satisfied; and

wherein the annul code comprises a second annul code in response to the condition predicate not being satisfied.

17. The processor of claim 16:

and further comprising a first register and a second register;

wherein the first annul code is stored in the first register; and

wherein the first annul code is stored in the second register.

18. The processor of claim 16:

and further comprising a register;

wherein the first annul code is stored in one-half of the register; and

wherein the second annul code is stored in one-half of the register.

19. The processor of claim 1:

wherein an annul instruction passing through the pipeline specifies an integer N; and

wherein the annul code is formed in response to the integer value N such that the circuitry for preventing prevents N successive instructions in the pipeline from altering the architected state.

96

20. The processor of claim 1:

and further comprising a register;

wherein the register stores the annul code which comprises a set of bits having a first logical value and a set of bits having a second logical value;

wherein the annul code is loaded in response to an instruction having a condition predicate;

wherein the circuitry for preventing prevents instructions corresponding to the bits having a first logical value from altering the architected state in response to the condition predicate being satisfied; and

wherein the circuitry for preventing prevents instructions corresponding to the bits having a second logical value from altering the architected state in response to the condition predicate not being satisfied.

21. The processor of claim 1 and further comprising circuitry for storing a portion of the annul code in response to receipt of an interrupt.

* * * * *